UNIVERSITY OF SASKATCHEWAN
DEPARTMENT OF COMPUTER SCIENCE

CMPT 250
Final Examination

3 hours **Closed Book** **April 18, 2006**
Marks

This exam is out of 150. Thus, you have 1 minute per mark with 30 minutes to spare. Use this in judging how much time to spend on a question.

---

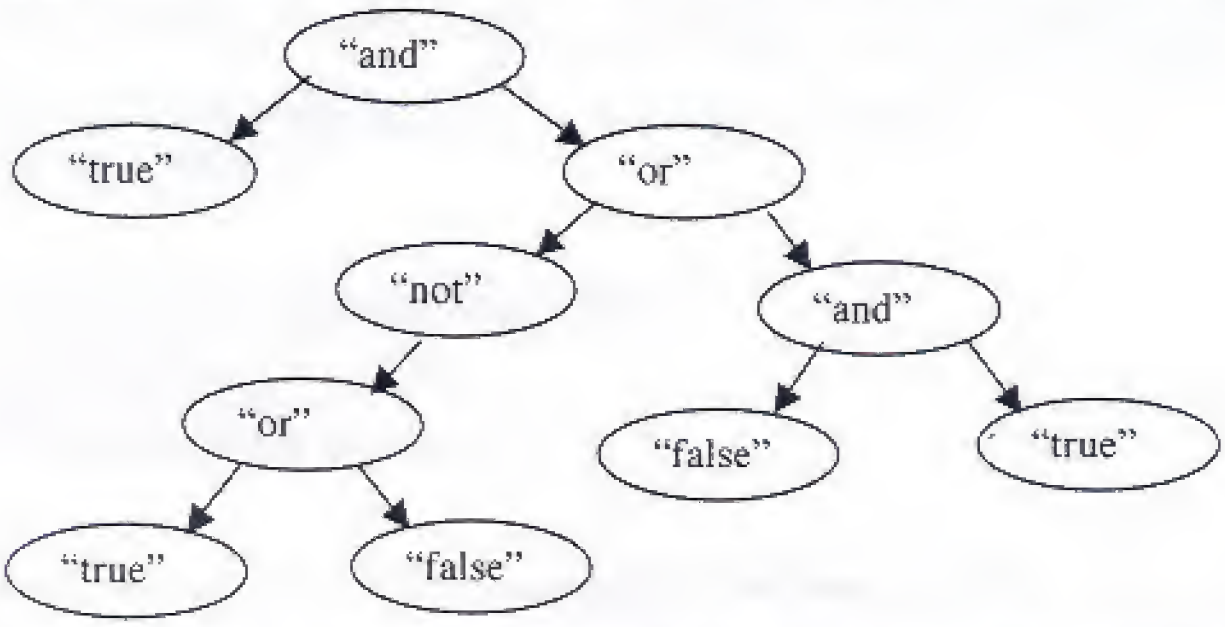
8 1. In assignment 9, you developed a routine to classify edges of a directed graph when a depth-first search is done. The edges were classified as follows:
tree edge: an edge of the depth-first tree
back edge: an edge to an ancestor in the tree of the current vertex
forward edge: an edge to a descendant in the tree of the current vertex
cross edge: an edge to a vertex in another branch of the tree (it will be a branch scanned earlier).
Give situations to be tested in order to do a through job of testing this routine. Note that since you are not given the code, this is black-box testing.

5 2. Consider the **directed** graph given below. Give the tree that is obtained by a **breadth-first** search of the graph **starting at vertex 4**. Also, for each non-tree edge that provides an alternate edge on a shortest path from vertex 4, draw it in the diagram with a dashed line.

1: 3, 9
2: 5 9 7
3: 6 8 5
4: 3 5 2
5: 8
6: 1, 5
7: 3
8: 1 6 5
9: 7 4 1

3. Two types of files that can be used to access an item/record by key value are
5 (a) a direct file
5 (b) a B-tree file
For **both** the file types, answer both of the following:
(i) How many file reads are needed to access an item by key? Justify your answer by describing what file reads are needed to access the item.
(ii) In designing the algorithms for the accessing of an item by key, what can be done to minimize the number of file reads needed?

22 4. In Eiffel, define a descendant of the class LINKED_SIMPLE_TREE_UOS[G] that stores STRINGs and has one added feature, a BOOLEAN function called *value*. The function is to return the BOOLEAN value obtained by evaluating the current tree. In each leaf, the STRING will be either the string "true" or the string "false", representing the corresponding BOOLEAN value. In an interior node, the STRING will be either "and", "or" or "not", representing the corresponding BOOLEAN operation that is to be applied to the subtree values. If an interior node has the string "not", then the left subtree stores a BOOLEAN tree and the right subtree is empty. You can assume that the current tree has valid values in its leaves and interior nodes. The function *value* should be designed to only handle non-empty trees. As an example, the following tree has value *False*.



For reference, the main public features of LINKED_SIMPLE_TREE_UOS[G] are

- make
- is_empty : BOOLEAN
- initialize (lt : like Current; r : G; rt : like Current)
- root_item : G
- root_left_subtree : like Current
- root_right_subtree : like Current
- out : STRING

13 5. In class and in the text, the axiomatic ADT was given for a binary tree type T storing items of type G. The build operations were
make - yields an empty tree
initialize(lt, r, rt) – yields a tree with *r* at the root, *lt* for its left subtree, and *rt* for its right subtree
Give the signatures, preconditions, and axioms to define the following operations for an ordered binary tree:
is_empty – a Boolean function to test whether the tree is empty
max – a function that yields the largest item in the tree
insert(i), i an item – yields a tree that is the same as the original tree except that *i* has been inserted into it
These last three operations should assume that the items are comparable, and the target tree is ordered. Also, the tree that *insert* yields must be ordered.

12 6. Consider a 2-3 tree with one variation. As normal, there are two types of nodes, interior nodes and leaf nodes. Normally, a leaf node stores one item and its key. For this question, assume that a leaf node stores *b* items and their keys, where *b* is a positive integer value. Give a general algorithm for insertion into this tree. This algorithm should include the handling of all the cases that are needed, but need not give the details for common tasks. Also include a discussion of how polymorphism is used in this algorithm.

20 7. This question is about finding a special type of cycle by means of a depth-first search of a directed graph. When a depth-first search is done in a directed graph, a (back) edge from some vertex (other than the start vertex) back to the start vertex forms a cycle. You are to give a detailed algorithm to determine the length of the longest such cycle.

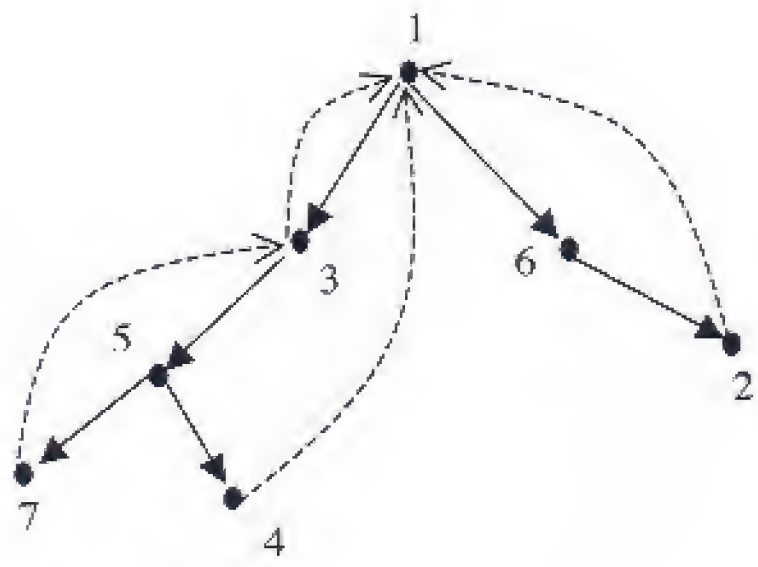
For your detailed algorithm, you can use the following notation:

```
For each vertex v
        . . .

        For each vertex u adjacent from vertex w
                . . .
or
        For each neighbor w of vertex v
                . . .
```

If you wish, you can give Eiffel code instead of a detailed algorithm.

Here is an example of a depth-first tree that shows the back edges found during a search.



Edge (3, 1) defines a length 2 cycle of interest. Edge (2, 1) defines a length 3 cycle, and edge (4, 1) defines a length 4 cycle. Edge (7,3) defines a cycle, but it is not the type that we are considering as the edge does not go back to the start. The cycles to be considered follow a tree path from the start to some vertex and then have an edge from this vertex back to the start. Only these cycles are to be considered.

10 8. Prove by induction on n that the number of strings of length less than or equal to n from an alphabet A of size m is $(m^{n+1} - 1)/(m-1)$. In other words, for a given value of n, we want the count of the number of possible strings where the length of a string must be between 0 and n (inclusive), and the characters are chosen from the set A of size m. For example, if n=2 and A={a,b}, the possible strings are

^, a, b, aa, ab, ba, and bb

where the empty string is represented by ^. In this case, the count is 7.

50 9. A rental system is used by an agent to track the allocation of rental items for the temporary use of customers. A rental item is allocated to a customer for certain duration of time called a rental period. A rental period will be a multiple of some time unit such as hours, day, and so on. Different items will have different fees (including zero fee) per time unit.

Rental items can be reserved for rental periods in the future. Reservations can be cancelled before the rental period starts. The rental system may also provide waiting lists for items that are allocated or reserved, as there is the possibility of a cancellation freeing up the item.

Payment for one or more rental items can be in cash, by credit card, or by cheque. A credit card payment has an associated authorization process with the Credit Card Company. An authorization request is made to the Credit Card Company for a certain amount and the company either denies or authorizes the request.

Once the rental period has ended, two actions are possible. The item must be returned, at which time the fee for the rental will be due, or the rental agreement can be renewed. An item should not be renewed if there is a waiting list or a future reservation that might overlap with the renewed rental period. If the item is neither returned nor renewed, the customer will be liable for overdue charges.

For each rental, customer information (such as an address where the customer can be located) must be maintained so as to be able to track down the item if it is needed urgently during the rental period or if the item is not returned or renewed after a suitable grace period.

You are to design an object-oriented software system for this problem. The deliverables consist of the following:

(a) The use case diagram with direct and indirect actors. If any use cases are not obvious from their names, briefly describe each one.

(b) Three interaction (sequence or collaboration) diagrams that include the following two use cases
- the rental of an item by a customer
- return of a rented item by a customer

(c) An overall class diagram for your system, i.e., include all classes and their associated relationships for your system. Make it as inclusive as you can (don't just include the classes and relationships from part (b)). Note that subsystems are not required, but can be used.

(d) Any inheritance diagrams with attributes and routines shown. If the class for rental item is not in one of these diagrams, give a class diagram showing its attributes and routines.

(e) Identify the containers in the system. For each container that is to be stored as a file, give its structure (sequential, direct, or B-tree) and its primary key. If it is to be stored in memory, describe the data structure to be used.

Total 150 **The end**